Salient Movies

by

Karrie Karahalios

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering

and

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author .....
Department of Electrical Engineering and Computer Science
May 26, 1995

Certified by .....
Andrew B. Lippman
Associate Director, MIT Media Laboratory
Thesis Supervisor

Accepted by .....
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Theses

Salient Movies

by

Karrie Karahalios

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 1995, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science in Electrical Engineering
and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis introduces a method for visualizing the spatial composition of a movie. It transforms time into space by classifying frames from a movie sequence into clusters of spatial similarity.

The approach used is derived from stochastic process theory. A training set of the data is taken from the movie. Principal component analysis is used to define the space of the frame clusters. Every frame from the movie is then clustered with respect to its distance from the designated frames in the designated space.

Movies are made up of many frames. The huge size of the data set motivates us to approach the problem in this manner. It also includes the added benefit of compression. In using this approach, we will actually be defining a new element of a movie. Elements of this nature are used to reassemble the movie. The result is a *Salient Movie.*

Thesis Supervisor: Andrew B. Lippman
Title: Associate Director, MIT Media Laboratory

This work was supported by contracts from the Television of Tomorrow consortia.

Acknowledgments

No work is a monolith; neither is any author. A large number of people have contributed to this thesis. I owe them all my gratitude.

Andy Lippman, for his criticism, his support in helping me find my way throughout this turbulent year, and for giving me the opportunity to be here.

Walter Bender, Michael Massey, and Costa Sapuntzakis for helping me understand the Salient Stills process.

Sandy Pentland and Thad Starner for answering all my questions regarding Eigenfaces.

Shawn Becker, Dan Gruhl, Nuno Vasconcelos, Roger Kermode, and Ed Chalom for offering suggestions to my problems, even at the most inopportune moments.

Henry Holtzman for keeping the system up so that I could retrieve my data and for getting me my own terminal.

Celia Shneider for always finding time to schedule me in.

Klee Dienes and Michelle McDonald for sharing with me the skill of debugging in the early hours of the morning.

Frank Kao and Eng Khor, for their help in uncovering creative ways to digitize.

Martin Szummer for the late night discussions on shot parsing, his interest in this work, his generosity, and for always smiling.

Jill Kliger for keeping me company late at night and for providing moral support and bagels.

Teresa, Marcie, and Tracy, my closest friends and roommates, for waking me up in the mornings, and for helping me maintain my sanity.

Finally, I would like to thank my mother, father, and my brother, Tasos - for always believing that I know what I'm doing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

When watching a movie today, one has the option of going to a movie theater, driving to the local video store, or turning on the TV and watching whatever might be on. In the not too distant future, video on demand will be commonplace and one will be able to access any movie through their digital television. High-speed networks will connect the viewer to vast databases of film media. One will no longer need to accept only what is broadcast on television as is the case now. With thousands of movies to choose from at any time, the new problem arises of how to make a selection. This motivates the need for the development of movie browsing tools.

There are currently two very different trends in movie browsing. The first involves the tagging of the characteristics of a movie, such as the actors, theme, and setting. The user can then perform a lexical query on this key-word database in order to browse through movies. The disadvantage of this technique is that describing a movie with text in this manner is too constraining. One will know the characters, but is limited in the knowledge of their interactions. Another problem with this method is that it requires a preliminary cataloging of the movie. Someone must manually label the significant events in all the movies.

The second method is to *download* or retrieve suggested sequences from a movie. Television and movie theaters currently use this notion of a trailer to portray the

highlights of a movie. A drawback to this method is that downloading and viewing one to two minute splices from many movies can be tedious and time consuming.

The method developed here allows one to visually browse a movie by creating a visual catalog of the movie events. Still images are generated from the spatial content of the movie, transforming it into its spatial domain by capturing its salient features. Unlike the key-word database, this is a stand-alone operation and does not require human intervention. Furthermore, it is a condensed representation of the entire movie and not just select highlights. This allows for easy browsing and also preserves the tempo and mood of the movie.

Video on demand as described earlier is not yet available. It will, however, exist in the coming future as networks, storage, and digital television grow evermore capable of supporting it. With the expanding banks of information, techniques for browsing and encoding data are essential. This thesis addresses the issue of browsing dynamic visual databases. In particular, I suggest a method of extracting features from a movie stream that will allow for the categorization, browsing, and filtering of a movie based on its texture and rhythm.

1.1 Domain

> *... Film is a magnificent and dangerous weapon if it is wielded by a free mind. It is the finest instrument we know for expressing the world of dreams, of feeling, of instinct.*[1]

1.1.1 The Making of Movies

The first ten years of film brought forth the reaction of wonder. The fascination lied in the creation of the motion. Movies were a sequence of static frames, where the events followed one another without any breaks. Characters essentially entered the

[1] Jean Claude Carriere, *The Secret Language of Film.*

frame and exited the frame as if it were a stage. Not until the use of editing did the making of movies change form. Editing created a new *language* for filmmakers.

Consider an image of a man in a dark room leaning against his window, staring passively at a storm outside. Following this, is an image of a woman seated on an airplane on a runway, gazing at the drops of the storm settling on her window. The repeated interchanging of these two streams into one stream creates an image that the viewer can decipher. The mind formulates the content in the gaps of the stream continuously as one watches the movie. This interpolation is done either from common knowledge or personal experience. One might initially infer that these two people are thinking about each other. The formulations may change as the sequence progresses with the context of the movie, but the pieces are being constantly joined together such that they follow ones reason or logic. "Constructivist theory holds perceiving and thinking to be active meaning-making processes. Constructivist theory maintains that as a story unfolds, viewers unconsciously undergo a hypothesis forming, testing, and confirmation or reforming"[3].

Shaping multiple sequences into a cogent movie is a skill. The idea for a film originates with an artists vision. The screenwriter, director, cinematographer, and editor are responsible for bringing this concept into existence. Their task requires the skill of visualizing three-dimensional space and how it translates into the two-dimensional form of film.

The film captures a physical description of the domain of the movie as well as spatial information regarding the camera motion and object motion. Camera motions such as pans, tilts, and zooms are used to invoke emotional responses. A zoom on a face will make that face occupy more of the film area. It will emphasize that one person and the expression on their face. The locations, the dialogue, and the actors themselves each add a certain meaning and charm.

The manipulation of space is not complete after the shooting process. More footage is taken than is usually used. The final grooming is done during the editing

phase. Here, the editor molds the movie in its two-dimensional form. The length of a shot is determined as well as the transition from one shot to the next. It is in this phase that the rhythm and fabric of a movie are created. Rapid cuts might be associated with excitement or culmination. Dissolves might indicate the passing of time or a dream-like state. These concepts are important to our understanding of movies. Capturing them is the essence of this work.

Work done by Edward Elliot while at the Interactive Cinema Group at the MIT Media Lab, explored methods of alternately portraying video with respect to the user and the editor[3]. He emphasizes the interactions between story making, story telling, and story viewing (see Figure 1-1).

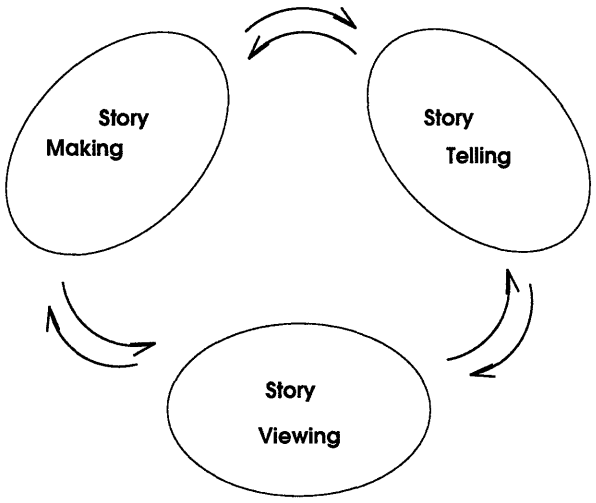


Figure 1-1: Story Activities

This thesis focuses primarily on *Story Viewing.* In order to understand and interpret video, we need to understand how it is put together.

1.1.2 Frames, Shots, and Scenes

In a manner of speaking, the atomic elements of a movie stream are the discrete frames used to compose it. These frames in turn make up a shot or a take. A shot is a sequence of frames where the run of the camera is not interrupted. A scene is comprised of one or more shots and illustrates an event or complete action. A movie, therefore, is made of one or more scenes.

These narrative elements of the movie not only advance the story itself, but they also direct our experience of it. A frame standing alone has a different quality from two successive frames. The lone frame represents an instantaneous moment in time, whereas two successive frames may exhibit movement and a lapse in time. Similarly, when two separate shots are joined, a novel concept is created which is unlike that of each of the distinct shots respectively. Pursuing this analogy, we can see how two scenes spliced together affect the movement in a movie stream. A sequence consisting of four shots is shown in Figure 1-2.

The human eye can parse this sequence into shots without any difficulty. It can also detect the spatial discontinuity between successive shots. Yet when watching a movie, one has the ability to piece together these fundamental cinematic elements into one continuous and rationally cohesive theme. This phenomenon also has a memory factor, in that with the passing of time, one also has the ability to remember key moments and settings in a film. This suggests that perhaps some insight can be gained by disassembling the movie stream into basic cinematic elements and logically building it again in an attempt to bring forth the essential images of the movie.

1.2 The Problem

The problem presents three technological challenges:

- The decomposition of the movie into simpler elements.

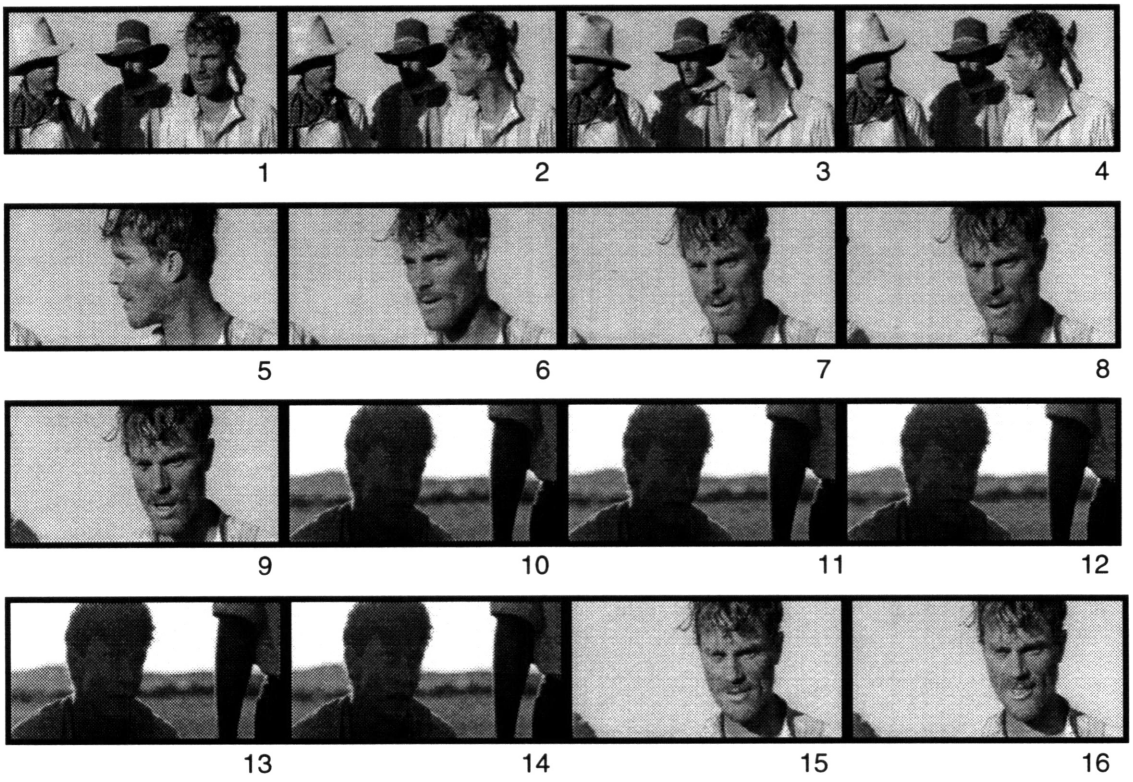


Figure 1-2: Example of Shot Juxtaposition

- Creating new building blocks out of the constituent elements.
- Recreating the movie from this new set of building blocks.

A film stream can be segmented into frames, shots, scenes, or any other well defined classification. We must first choose what to use as our primary element in the separation of the movie. Different elements cater to different movies. Alfred Hitchcock's film *Rope* contains three hidden edits. To the human eye, it appears to be one long shot. Sequences with extensive zooms and fade-ins, should also be considered. What element would best describe such situations?

Parsing the movie into frames is trivial. We would be done before we even started and would gain no further information. Parsing the movie into shots is feasible and relatively time efficient. Scene parsing is possible, however, it is currently not very time efficient. Eventually, the parsing of scenes should be pursued, but for now, we will concentrate on the segmentation of shot-like elements.

The next concern is how to determine which frames belong in which "shot". Most shot parsing algorithms to date use histogram level thresholds to differentiate shot cuts. This method introduced here relies on pattern recognition and classification techniques. We will classify the frames based on their content and not on dramatic changes in intensity level. By doing so, we are actually representing this new element as a class of frames, each of which have certain salient features in common.

Finally, we must reassemble the movie. The shot-like elements as defined by their classification, become the new *atomic* elements of our movie. Each class of moving images is transformed into one still image which outlines the spatiotemporal sequence of the respective entities in the class. This image does not represent one discrete moment in time, but is a structured representation of an entire event.

The problem we face is that "shots" are not so easily defined in a way that a machine can cluster them. In a movie, there is a sense of the space created by the arrangement of shots that is implicit in our viewing. We generally know when an event takes place in a single room or when it jumps to a different location. When a

camera pans, the background is constantly changing. Therefore, each frame will not have an identical background, and the characters will occupy different positions of frame area. Camera zooms pose the same dilemma. The characters gradually occupy more and more frame area, while temporarily eliminating the background.

A second aspect of the problem is the sheer amount of data involved. Even compressed, a movie is about a gigabyte per hour. Not only are all the bits in the frame different in each succeeding frame, but there are too many of them to simply use brute force.

1.3 Thesis Overview

The goal of this thesis is to transform time into space. We want to build a spatial representation of the movie that expresses some of what we would see were we to watch the movie sequentially.

This process involves the amalgam of several techniques. The classifying algorithm alone is a powerful system. It clusters all of the frames of a movie into a small number of groups that hopefully represent the various spaces used during the filming. Movies, however, do not often return to a site more than once during the course of the action. Therefore, the frames that form a cluster are not necessarily temporally proximate.

The approach for the classification was motivated by stochastic process theory. An *eigenspace* is described by a subset of the frames in the movie. The proposed algorithm suggests a method for classifying which relies on the variance between respective locations of frames in this well defined *eigenspace*. The scheme cannot be overly stringent, for the objects in the frames may be in motion. However, it must be accurate enough to create a point of comparison among the existing logical events.

Given this clustering, we return to the original film and examine each frame in turn. When the sequence of frames shifts from one cluster to another, we can call it a cut and mark it. We can thus do shot cut detection. For each cluster, we try to

build a single representative frame or still.

Creating the stills is a different problem in itself. Some method is necessary which can highlight the important features within a stream. Affine transformations are used for merging all the frames within a class into one frame. The technique involves recovering the camera motion from the given sequence using optical flow. The frames are then warped together on top of each other. In the rendering process of the single still image, changes in the focal length and field of view of the sequence are also taken into account.

Combined, these two techniques form *Salient Movies*. This process fragments the standard order and rhythm of the activity in the movie stream. The result is an array of spatiotemporal stills that can be used as a catalogue or browsing device.

This thesis begins by reviewing pertinent techniques for scene parsing, object and pattern recognition, and motion estimation. Chapter 3 discusses the *eigenframe* algorithm and the creation of the salient stills. An evaluation of these techniques is examined in Chapter 4, and Chapter 5, presents some examples of *Salient Movies*. Chapter 6 discusses points of further interest. Finally, Chapter 7, concludes this thesis with a brief discussion.

Chapter 2

Background

2.1 Parsing Techniques

2.1.1 Traditional Approaches

Unless cuts are recorded during the actual filming and editing process, other methods of shot declaration must be used. Research in the field of shot and scene segmentation is not new. In the past, the combined use of histograms and statistical analysis prevailed. These methods are still being used today in commercial products and research because of their robustness and near real-time capabilities. Recently, however, methods which utilize data in compressed form have also been examined.

Most existing parsers utilize histograms to detect scene cuts. I will describe one such model that was recently implemented by Martin Szummer at the Media Lab [9]. It consists of a three parts as shown in the flow chart in Figure 2-1.

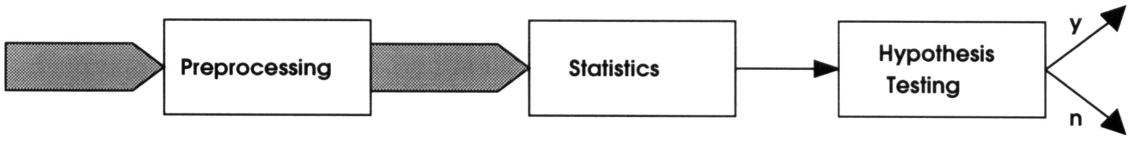


Figure 2-1: Color Transition Based Shot Cut Model

Preprocessing

Preprocessing manipulations are widely used to describe an image with less information. Common examples of preprocessing include, subsampling and filtering. Since such computations can also be costly, they should ideally be performed before the task at hand to avoid extra delays in near real-time processes.

Statistics

The next step involves measurements on the color levels of the new images. Several computations used alone or combined make up the statistics module of the flow chart in Figure 2-1. These measurements are computed between pairs of successive frames. Table 2.1 contains some of the most common shot cut detection methods [4].

Table 2.1: Common Shot Cut Measurement Formulas

Name	Formula
Difference	$\forall_{x,y} I(x,y,t+1) - I(x,y,t)$
Division	$\forall_{x,y} I(x,y,t+1) \div I(x,y,t)$
Gradient	$\forall_{x,y} I(x+1,y,t) - I(x,y,t)$
Threshold	$\forall_{x,y} B(x,y,t) = 1$ if $I(x,y,t) \geq T$
Mean	$\frac{\sum_{x,y} I(x,y,t)}{N}$
Variance	$\sum_{x,y}(I(x,y,t) - \frac{\sum_{x,y} I(x,y,t)}{N})^2$
Histogram Difference	$\lvert H_t(i) - H_{t+1}(i)\rvert$
Histogram χ^2	$\sum_{i=0}^{G} \frac{(H_t(i)-H_{t+1}(i))^2}{H_{t+1}(i)}$

$H_t(i)$ represents the number of pixels in *ith* bucket at time t. Classification may be improved by utilizing the information from color in the given formulas. We must keep in mind that the performance of these methods is not independent of the prefiltering operations.

Hypothesis Testing

Several different choices also exist for the Hypothesis testing module. This portion requires a clear cut "yes" or "no" response. Given a logical condition, the numerical result from the statistical methods is compared to a threshold value. The logical condition might be a distance, a derivative or a test specific to the desired response.

2.1.2 Parsing of Video in Compressed Form

A logical extension to scene and shot parsing is the segmentation of video in compressed MPEG and JPEG form. Such methods rely on the correlation of Discrete Cosine Transformation(DCT) coefficients of consecutive frames[14]. The process requires multiple passes. The first pass determines potential shot cuts, and the resulting passes perform more rigorous detection algorithms. As with most parsing techniques, false positives arise when combinations of zooms, translations, or rotations are present.

2.1.3 Parsing Video with Audio

Audio is a key element in movies. It possesses a great deal of information with regards to the content and action of the movie. Audio can be used to detect explosions, screaming, and other distinct sound effects. In a sequence which contains a shot outdoors in a crowded street in Manhattan, followed by a shot in a quiet apartment, audio can be used effectively in determining shot transitions.

2.1.4 Commercial Parsers

Dubner International, Inc. produces a commercially available cut detection package called *Scene Stealer*[2]. Its specific use is for the logging and cataloging of videotape. Fade ins, fade outs, zooms, pans, and tilts are of no concern in the logging of such raw footage. Thirty frames per second are used in the processing and the *key* frames representing cuts are stored for logging purposes. The system works amazingly well

for its simplicity[1].

The algorithm follows the traditional approach. Most of the system is integrated on a circuit board installed in a PC, while the hypothesis testing resides in software. The images are first passed through a simple three pole low-pass filter with a cutoff point of about 60 kHz. Six samples are then taken per line per field resulting in about 1,440 samples per field. Essentially gray-level values are subtracted from the corresponding values two fields earlier[2]. All the absolute values of the differences are added together. This signature is compared the the previous two fields and the later two fields. If it crosses a threshold value, then a cut is declared.

2.1.5 General Parsing Observations

The methods described above work extremely well. Deciding which method to use depends on the problem at hand. As a general rule, these methods require a high frame rate, especially when attempting to parse fast-paced action such as MTV videos, explosions, commercials, and cartoons. Real time manipulations with user interaction becomes feasible when heavy preprocessing is performed and a clever selection of test points is made. For example, if using a histogram approach which requires exhaustive computation, a common technique is to choose few statistically relevant points rather than the entire data set (see Figure 2-2)[9].

2.2 Object and Pattern Recognition

In many ways, segmenting shots from a video stream is analogous to segmenting an object from an image. When recovering objects by edge-detection or line-finding, we are essentially seeking boundaries within the same medium. Edge detection approaches the problem from one direction, whereas line-detection merges from each

[1]The accuracy is close to 100% (98%) for clear cuts.
[2]This is the difference method listed in Table 2.1

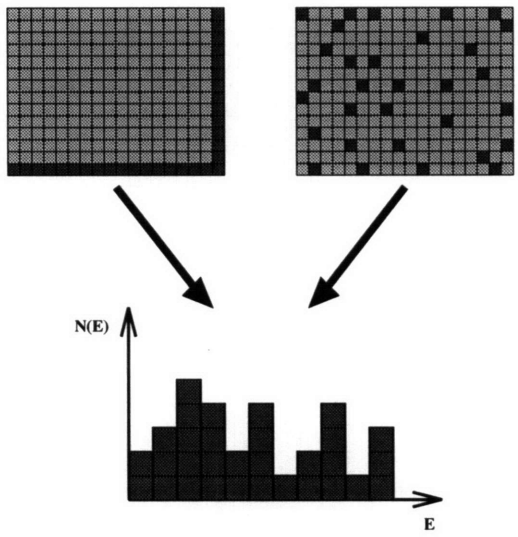


Figure 2-2: Example of using only 2 borders and statistically chosen points within a frame for the purpose of statistical computation.

side. Histogram thresholds provide information as well with regard to boundaries within an image.

Object recognition techniques, however, can also be pursued from image to image in the aid of shot parsing. For example, if the same object exists in a sequence of frames, it is extremely likely that that object lies in the event space of a shot.

The approach used in this thesis is that of pattern recognition. Pattern recognition deals specifically with the classification of patterns of interest. These patterns can be characters, biological cells, electronic signals, or any other object that one may wish to classify [11]. We will be classifying frames of a movie and treating them as patterns.

2.2.1 Eigenfaces

Eigenfaces refers to a technique used for face recognition[8, 12, 13]. Principal component analysis is used to handle the recognition. This method does not require a three-dimensional recovery of the face, but rather uses the two-dimensional charac-

teristics of the face for the computations. The most uncorrelated features of the faces are used to discriminate between faces. These include the eyes, noses, and mouths.

The face image is projected onto *eigenspace* which is made up of the known training set of faces. This projection characterizes each face as a weighted sum of the calculated *eigenfaces.*

It is a practical technique that should work assuming the faces have approximately the same orientation, i.e. a mug-shot type stance. Since the lighting and orientation were constant at all times, these facial features were in relatively the same position in each respective frame. Rotated faces and translated faces are not easily recognized.

2.3 Visualizing Time and Space

So far, I have discussed methods of parsing, segmenting, and classifying. This section is devoted to the combination of frames. When working with a large number of images, we are posed with the problem of how to present them at once.

A series of frames can be combined and viewed in a variety of forms. We are most familiar with seeing frames displayed on top of each other in a rapid succession. This is a localized method with no trace of the past, yet with a *real* effect. It creates an abstraction for the user that eliminates the projector and focuses only on the viewing screen.

To browse through a film without actually letting it "roll" before you, a logical extension would be to watch a condensed form of the film. In the sense that a frame is analogous to a word, we can read frames from left to right similar to the frame sequence in Figure 1-2. For a shorter description, we can view only frames at equal intervals (see Figure 2-3). Alternatively, a frame from each shot may be displayed or perhaps the first and last frames of each shot. David Small at the MIT Media Lab's Visual Language Workshop has incorporated a temporal link to the frame display by placing the key frames from each shot on a timeline. This is not a bad idea, for much

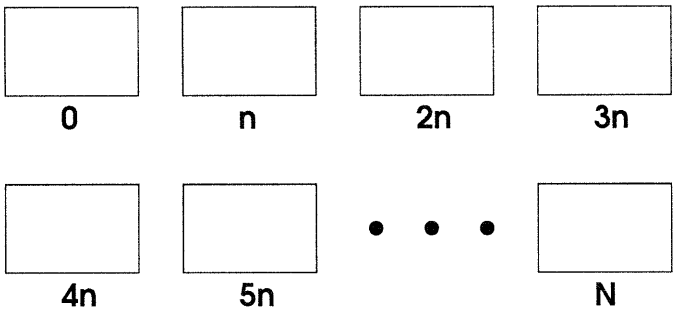


Figure 2-3: Viewing Frames at Equal Intervals

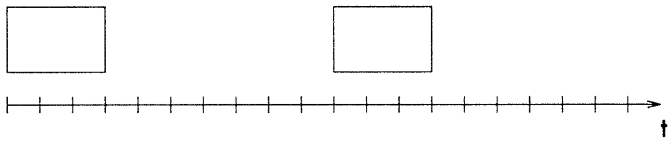


Figure 2-4: Viewing Frames on a Timeline

of the information in successive frames is redundant (see Figure 2-4). Edward Elliot at the Media Lab's Interactive Cinema Group, has subsectioned the frames to show time as depth in his video streamer (see Figure 2-5).

2.4 Salient Stills

The use of Salient Stills, developed by Laura Teodosio[10], takes advantage of the redundancy in frames. A frame standing alone captures a two-dimensional representation of a three-dimensional world at an instantaneous moment in time. Displaying two consecutive frames establishes a time frame while sacrificing spatial continuity. The probability that two adjacent frames in a shot sequence possess much of the same information is very high. The Salient Still approach creates a composite of these two similar frames. The same object in each frame is warped into one object on one frame (see Figure 2-6).

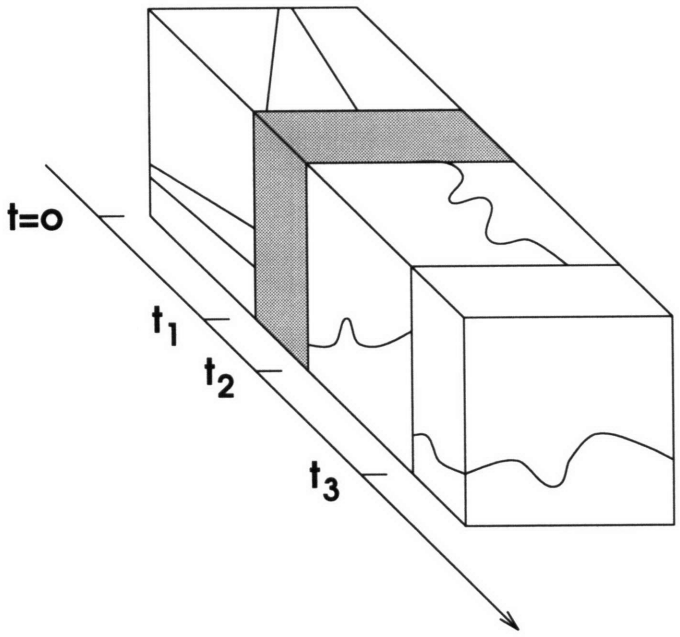


Figure 2-5: Video Streamer

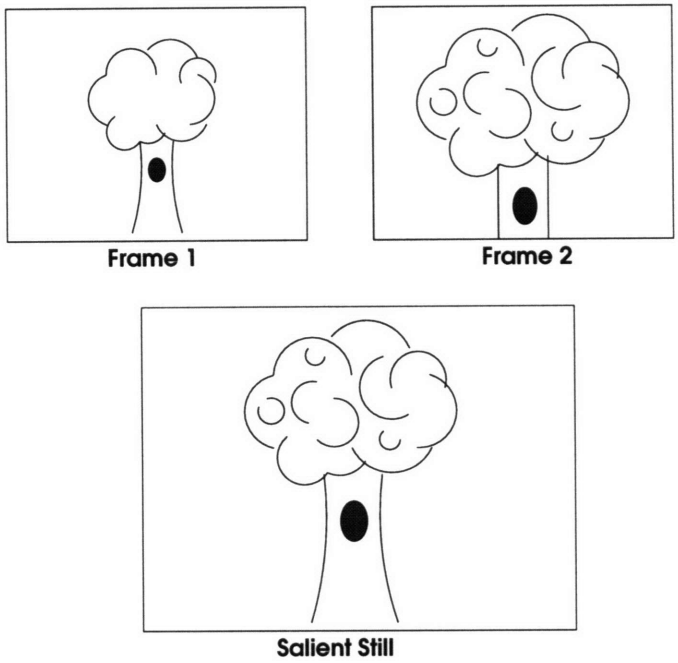


Figure 2-6: Example of Salient Still Composite

2.4.1 Process

There are three main portions in the Salient Still methodology. These are the estimation of the optical flow, the warping using the affine coefficients, and the final filtering of the image data. A block diagram of the still process is shown in Figure 2-7.

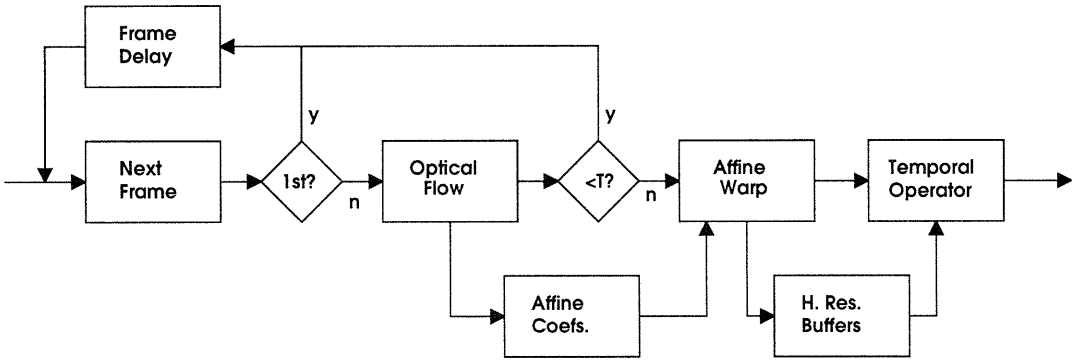


Figure 2-7: Salient Still Block Diagram

Optical Flow

The first step in the Salient Still process is the calculation of the optical flow. Optical flow is the perceived motion of the image intensity[5]. It is modeled by the equation:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.1}$$

An affine transformation is used to estimate the optical flow. In a six parameter model, the velocity, $v(x, y)$ is represented by the equations:

$$v_x(x, y) = a_x + b_x x + c_x y \tag{2.2}$$

$$v_y(x, y) = a_y + b_y x + c_y y \tag{2.3}$$

where a_x and a_y are pixel translations in the x and y directions, respectively. b_x and c_y are percentage scaling factors in the x and y directions, respectively, and c_x and b_y

are percentage rotation factors in the x and y directions. Whereas optical flow was non-linear, the estimation has become a linear function. An explanation of how the affine parameters are determined can be found in Appendix A.

Warping

We want to warp frame t onto frame $t+1$, frame $t+1$ onto frame $t+2$ and so on. A frame is chosen as the starting point, and then affine transformations are summed towards that frame. The affine transformations are summed using the equations:

$$ax_{new} = ax_1 \times bx_2 + ay_1 \times cx_2 + ax_2 \tag{2.4}$$

$$ay_{new} = ax_1 \times by_2 + ay_1 \times cy_2 + ay_2 \tag{2.5}$$

$$bx_{new} = bx_1 \times bx_2 + by_1 \times cx_2 \tag{2.6}$$

$$by_{new} = bx_1 \times by_2 + by_1 \times cy_2 \tag{2.7}$$

$$cx_{new} = bx_2 \times cx_1 + cx_2 \times cy_1 \tag{2.8}$$

$$cy_{new} = by_2 \times cx_1 + cy_1 \times cy_2 \tag{2.9}$$

All the images are now defined with respect to one new global image.

Filtering

The final picture has different effects depending on the filter used. Many different operations can be performed on the pixels in their corresponding locations in the final buffer. A *mean* filter essentially averages the pixels in place. This operation, however, tends to leave ghost-like effects behind. The *last* operation returns the pixel values corresponding the the last frame. Similarly, the *first* operation returns the pixel values corresponding to the first frame. Of all the operations, currently the *median* operation gives the best results. The median filter also tends to eliminate

transitory objects from the sequence such as moving cars or moving people.

2.4.2 Performance

An added benefit of using this method is the increased resolution of the image in the regions of interest. With a variety of frames from close-ups to pans, the final picture will have more information in some regions than others. If there is a zoom of a face, that face will have increased resolution in the final picture. Since cameras tend to focus on the important aspects of an event, the important events will be clearer.

Salient Stills work amazingly well for certain situations. There are some situations where the estimation is less than perfect. This might occur when there exists a change in the vanishing point, when there are extreme changes in lighting[3], or when a moving object occupies a large portion of the window of estimation[4]. The method, however, is extremely flexible and works well in enough situations such that it can be applied easily.

[3]This is because only the luminance values are used.
[4]Most of the pixels should be well correlated.

Chapter 3

Classification using Feature Extraction

3.1 Introduction

To this point, it may seem as if I have been using the terms *parse*, *segment* and *classify* interchangeably. Parsing and segmenting refer to the act of separating an entity into component parts, which in this case are elements of a movie. What I am proposing is the classification of each frame of the movie into a correlated cluster. This cluster will then define a *shot-like* element. The problem now is how to classify the frames into natural groupings.

Several different parsing techniques have thus far been discussed. Most of them rely on pairwise intensity comparisons of successive frames. These comparisons are performed in spatially localized areas of the frame regardless of the frame content. The following method introduces a method for extracting features in an image. We will classify the frames based on the variance of these features in each frame.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) describes an entire system with several linear combinations of the original variables [6]. This is done through the use of the variance-covariance structure of the system. It is sometimes also referred to as the *Karhunen-Loeve* expansion.

Geometrically, a new coordinate system is created where the axes represent the directions of maximum variability. The principal components are derived from the covariance matrix. To illustrate this, let the random variable $\mathbf{X^T} = [\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}, \ldots, \mathbf{X_M}]$ with covariance matrix $\mathbf{C}$ and eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_M \geq 0$. Consider the equations:

$$\begin{aligned} Y_1 &= \mathbf{e_1^T X} = e_{11}X_1 + e_{21}X_2 + \cdots + e_{p1}X_p \\ Y_2 &= \mathbf{e_2^T X} = e_{12}X_1 + e_{22}X_2 + \cdots + e_{p2}X_p \\ &\vdots \\ Y_p &= \mathbf{e_p^T X} = e_{1p}X_1 + e_{2p}X_2 + \cdots + e_{pp}X_p \end{aligned} \tag{3.1}$$

These are linear combinations of $\mathbf{X}$. The covariance and variance are shown in the following equations:

$$Var(Y_i) = e_i^T \mathbf{C} e_i \quad i = 1, 2, \ldots, p \tag{3.2}$$

$$Cov(Y_i, Y_k) = e_i^T \mathbf{C} e_k \quad i, k = 1, 2, \ldots, p \tag{3.3}$$

Of the equations in 3.1, the principal components are those with the higher variances. The coefficient vectors, $\mathbf{e}$ should be restricted to unit length to eliminate indeterminacy. Therefore, the *ith* principal component is the linear combination where

- $Var(\mathbf{e_i^T X})$ is maximized subject to $\mathbf{e_i^T e_i = 1}$
- $Cov(\mathbf{e_i^T X}, \mathbf{e_k^T X}) = 0$ for $k < i$

Assume the principal features are now, $\mathbf{Y_i} = \mathbf{e_i^T X}$, for i=1,2, ...,m. We still need to decide which of these to use. Let $\mathbf{u_i}$ represent the eigenvectors of the covariance matrix, $\mathbf{C}$. Since $\mathbf{e_i^T e_i = 1}$,

$$\max_{\mathbf{e} \neq \mathbf{0}} = \frac{\mathbf{e^T C e}}{\mathbf{e^T e}} = \lambda_1 = Var(\mathbf{Y_1}) \tag{3.4}$$

When $\mathbf{e} = \mathbf{u_1}$ we can see that the eigenvectors with the higher eigenvalues have the highest variance[1]. In the equation:

$$D = \frac{\lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_m} \tag{3.5}$$

D represents the fraction of total variance contributed by the *kth* principal component. In most cases, up to 90% of the the total space can be described using the first few principal components. Therefore, most of the information is retained in the principal components with the higher corresponding eigenvalues. These components can now be used in place of the original m variables.

3.3 The *Eigenframes*

In extending this method for describing frames within a movie we can take advantage of the befits of PCA:

- Large reduction in the amount of data.
- Understanding of the content.
- Encoding and decoding purposes.

[1] This is proven in [6].

3.3.1 Using PCA for the Classification of Images

The Representation of Frames

We have shown how we can represent a random m-dimensional vector, $\mathbf{X}$ with less than m components. We now wish to use this information for the purpose of classification.

The concept of *eigenframes* is an extension of the *eigenfaces* technique[8, 12, 13]. A typical movie has hundreds of thousands of frames. There exists a *frame space* which contains every frame of the movie. Each frame, $\mathbf{F}$, can be represented as a vector of picture elements. These picture elements are comprised of eight bit luminance values. Assume a two-dimensional frame has dimensions N by N. It can, therefore, be described with a vector of dimension N^2 in N^2-dimensional space.

Within this frame space we wish to find a subset which optimally describes the given frames. This new space, the *eigenspace* can be defined by the eigenvectors of the covariance matrix of the frames. We will call these eigenvectors the eigenframes of the movie. Each frame can now be approximated by its projection onto the eigenspace. This is identical to saying that we will define each frame to be a weighted sum of the eigenframes.

The Classification of Frames

A movie transmits information at 24 frames per second[2]. A two hour movie, therefore, contains 345,600 frames. The sheer magnitude of data increases the complexity of the problem.

In order to classify the frames, we begin by making an estimation as to the dimensions of the eigenspace. We have already shown that the dimensions of the eigenspace can be significantly smaller than the total frame space. We then extract that number of frames from the entire movie at equal intervals. These frames are used to form

[2]Television transmits at 30 frames per second or 60 fields per second.

a "training set"of images. It is on this set of images that we create the covariance matrix and ultimately create the eigenspace.

The eigenvalues and eigenvectors or eigenframes are derived from the covariance matrix. Each frame from the movie is then projected onto the eigenspace such that it is described as a weighted sum of the eigenframes. The eigenframes are then clustered with respect to their proximity to the known frames represented by the components of the training set.

Figure 3-1 illustrates a simple case of this classification. Suppose that the original frame space is three-dimensional and the corresponding eigenspace is two-dimensional as depicted with the shaded plane. The elliptical objects represent the known frames in the training set. The remaining frames are then projected onto this plane. They are classified into the group which is closest to their projection onto the eigenspace.

3.4 The Algorithm

The design of the system follows the block diagram in Figure 3-2.

3.4.1 Creating the Training Set

As we have already mentioned, the training set should consist of a subset of frames from the entire movie. Choosing the optimal training set is an altogether different problem that will be explored in Section 4.1. For now, we will span the movie and select frames at equal increments to compose our training set (see Figure 3-3). The number of frames in the training set, in a sense, defines the number of clusters. We can see already, that there exists the possibility of completely missing the detection of a scene if the increments used for acquiring the training set are very large.

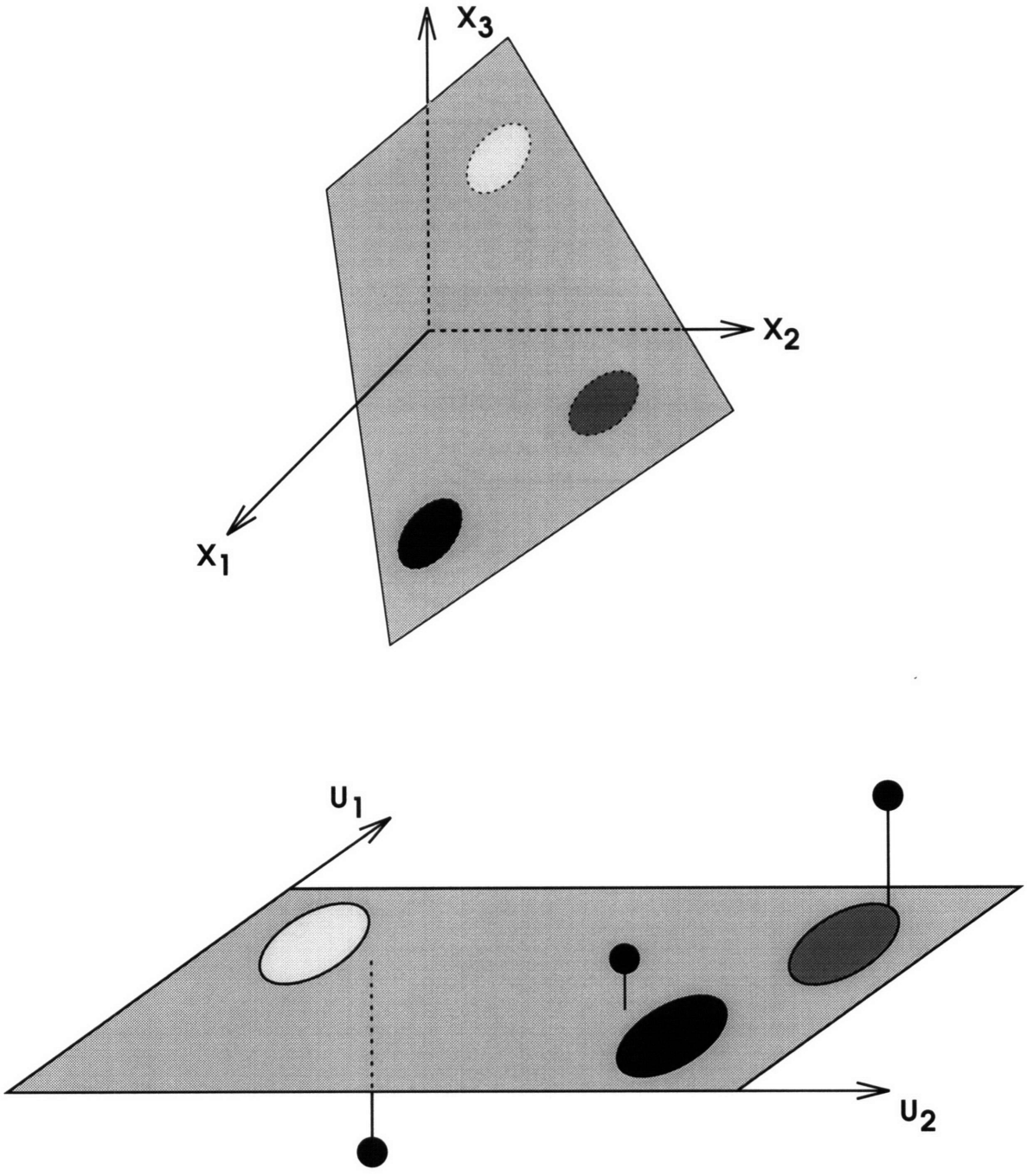


Figure 3-1: Simple Representation of Classification in *Eigenspace*

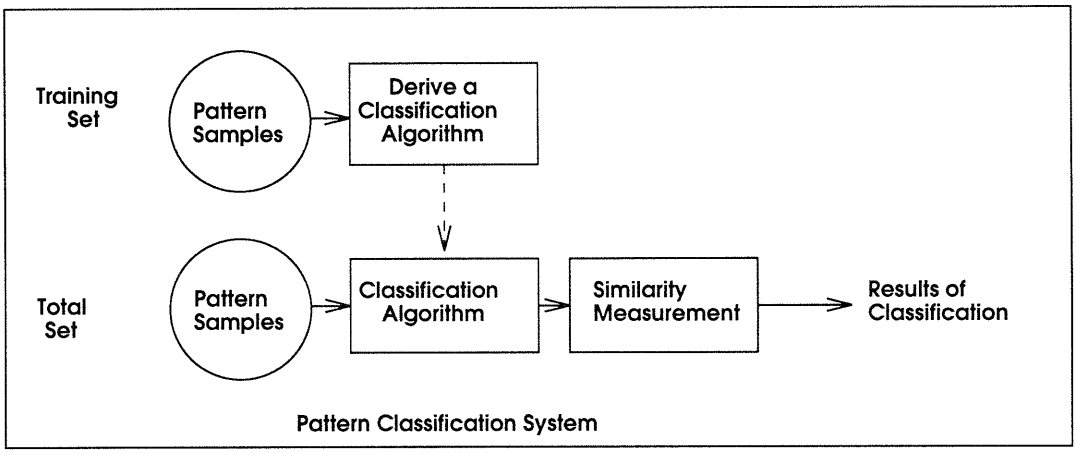


Figure 3-2: Block Diagram for the Classification System

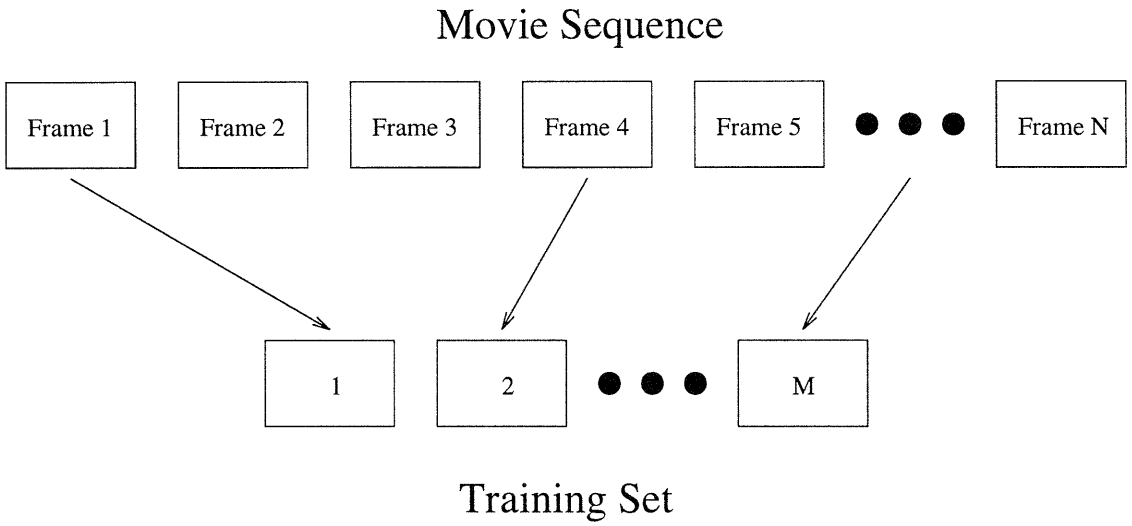


Figure 3-3: Acquisition of Training Set

3.4.2 Creating the Eigenframes

Consider a movie which is comprised of P frames. Assuming the frames of the movie have N rows and N columns. We will then represent each frame $\mathbf{F}$ as an array with N^2 elements in a predefined order. It can be seen that the frames need not be of dimension N by N. If the frames are of dimension R by N, then the frame is represented as an array of RN elements.

The training set is composed of images $T_1, T_2, T_3, \ldots, T_M$ such that $M < P$. Averaging all the frames in the training set, we get the sample mean, $\mathbf{\Psi} = \frac{1}{M}\sum_{i=1}^{M} T_i$. $\Phi_i = T_i - \mathbf{\Psi}$ defines the difference of the average from each member of the training set. A sample sequence is shown in Figure 3-4. The training set of this sequence consists of frame 1 and frame 5. The mean frame of the training set is pictured in Figure 3-5.

Since the set of Φ vectors are very large, it is beneficial to analyze these images in terms of features. The covariance matrix of the training set is represented by

$$C = \frac{1}{M}\sum_{i=1}^{M} \Phi_i \Phi_i^T = QQ_T \tag{3.6}$$

where $Q = [\Phi_1, \Phi_2, \ldots, \Phi_M]$. Our original images had N^2 elements. Therefore, the covariance matrix, C has dimensions N^2 by N^2 and N^2 eigenvalues and eigenvectors. Solving for N^2 eigenvalues and eigenvectors, is a computationally expensive task given the size of the images. Since we have M images in our image space, where $M < N^2$, we know that at most M eigenvectors are necessary to describe the eigenspace[3]. The eigenvalues and eigenvectors of C must satisfy the equation:

$$C\mathbf{u_i} = QQ^T\mathbf{u_i} = \lambda_i \mathbf{u_i} \tag{3.7}$$

where $\mathbf{u_i}$ represents the eigenvectors and λ_i represents the eigenvalues of C.

[3]There actually exit $M - 1$ valid eigenvectors. The Mth eigenvector contains all zeros.

Figure 3-4: A Seven Frame Movie Sequence

Figure 3-5: The Average of the Training Set

Q^TQ, however, is an M by M matrix whose eigenvalues and eigenvectors satisfy the equation:

$$Q_TQ\mathbf{v_i} = \mu_i\mathbf{v_i} \tag{3.8}$$

where $\mathbf{v_i}$ and μ_i are the eigenvectors and eigenvalues of Q^TQ, respectively.

Multiplying both sides by Q gives us

$$QQ_TQ\mathbf{v_i} = \mu_iQ\mathbf{v_i} \tag{3.9}$$

Comparing this equation to equation 3.7, we can see that $\mathbf{u_i} = Q\mathbf{v_i}$ are the eigenvectors of the covariance matrix $C = QQ^T$. These eigenvectors or eigenframes are then ordered with respect to decreasing magnitude of their eigenvalues. The eigenframes with the higher eigenvalues represent the axes with the directions of maximum variability. It is even possible to retain only $M' < M$ eigenvectors for further compression. Sometimes this is advantageous, for some of the lower eigenvalues may form features representing noise. It is in this new coordinate system that we will determine the similarity between frames and their respective clustering.

3.4.3 Mapping the Frames onto the *Eigenspace*

Each of the frames in the movie must now be projected onto the eigenspace. It is here where we will define the elements of the movie.

For each frame, $\mathbf{F_i}$, the features acquired by projecting that frame onto each of the eigenframes are represented by:

$$f_{i,j} = \mathbf{u_j^T}(\mathbf{F_i} - \mathbf{\Psi}) \tag{3.10}$$

where $i = 1, 2, 3, \ldots, P$ and $j = 1, 2, 3, \ldots, M'$. Therefore, the reconstructed frame can be described as,

$$\tilde{\mathbf{F}}_\mathbf{i} = f_{i1}\mathbf{u_1} + f_{i2}\mathbf{u_2} + \cdots + f_{iM'}\mathbf{u_{M'}} \tag{3.11}$$

3.4.4 Similarity Measures for Classification

Figure 3-1 shows a representation of a two-dimensional eigenspace. When an image is projected onto this eigenspace, there are two distances we must consider. The first is the distance from the image to the eigenspace and the second is the distance between the respective projections in the eigenspace.

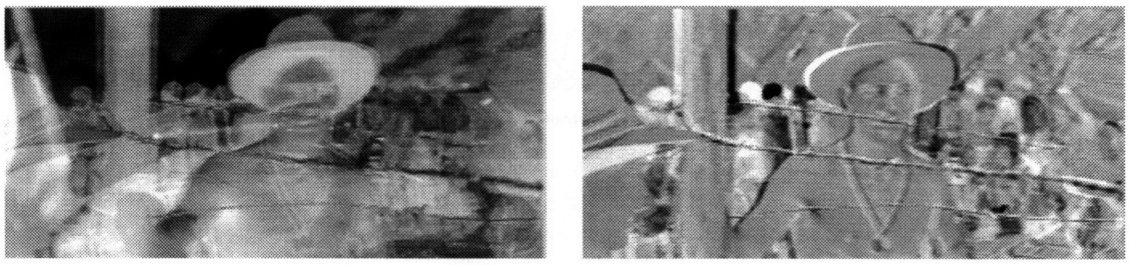
If a frame's distance from the eigenspace surpasses a threshold, then that frame is probably a lone frame and not part of a shot sequence. The distance between a frames projection and the known projections of the training set frames are used for the classification. The frame is classified into the group with the shortest distance between projections. The distance we use here is a Euclidean distance:

$$d(\mathbf{f_i}, \mathbf{f_j}) = [(\mathbf{f_i} - \mathbf{f_j})^T(\mathbf{f_i} - \mathbf{f_j})]^{\frac{1}{2}} \tag{3.12}$$

3.4.5 Examples of Sequence Classifications

Figure 3-6 displays the eigenframes of the sequence in Figure 3-4. The top frame displays the eigenframe for the case where the training set consists of frames {1,5}. The bottom two frames are the eigenframes for the training set which consists of frames {1, 4, 7}.

Case 1: M=M'=2



Case 2: M=M'=3

Figure 3-6: *Eigenframes:* The top picture is for a one-dimensional *eigenspace* and the bottom two frames are for a two-dimensional *eigenspace.*

The eigenspace in the first case is one-dimensional, thus the single eigenframe. In this space, two clusters were formed: Cluster 1 = {1,2,3,4} and Cluster 2 = {5,6,7,8}. In the second case, there are two eigenframes. In this case, three clusters were formed: Cluster 1 = {1, 2}, Cluster 2 = {3, 4}, and Cluster 3 = {5,6,7,8}.

3.5 Review of the Methodology

- Subsample the movie temporally.
- Filter and subsample the the frames of the movie.
- Create a training set from the frame set.
- Create the covariance matrix.
- Calculate the eigenvalues and eigenvectors of the covariance matrix.
- Order the eigenvectors with respect to decreasing eigenvalues.
- Determine how many eigenvectors to use.
- Project every frame of the movie onto the eigenspace and determine its distance with respect to the eigenspace and the known projections.

Chapter 4

Evaluation

This chapter begins by assessing the performance and limitations of the eigenframe clustering algorithm. In evaluating the ability of such decision making systems, the means are not always straightforward . We are analyzing a system which "breaks" a movie into smaller elements by clustering them. Of the known cinematic elements, this cluster most resembles a shot. We will, therefore empirically liken this element to a shot for comparison.

4.1 The Classification

The *ideal* classifier would have a member in the training set from each shot. Figure 4-1 is such a case. The clusters are denoted by the shading of the border of each frame. The training set consists of frames {0, 12, 24}.

To ensure that a frame from every shot is present in the training set, we would run into the "chicken and egg" dilemma. There are several factors then that influence the classification of shots.

- Selection of frames in the training set.
- Number of frames in the training set.

Figure 4-1: Simple Case of "Good" Clustering

- Camera motion.
- Character motion.

Table 4.1 examines how the number of frames in the training set affects the detection of shots. This may be a little misleading. Shot detection is quite good, however, many false positives are also found. This is due mostly to camera and character motion.

Table 4.1: Size of Training Set vs. Successful Detected Cuts

% of Frames used in Training Set	% of Found Shot Cuts
1	74
2.5	89
5	92
10	94

Consider the clustering in Figure 4-2. In this sequence, the camera is panning in such a manner that the foreground (the tree) is moving at a different speed from the background. Again, the training set consists of frames {0, 12, 24}.

Principal Component Analysis is not shift-invariant. A slight shift of an object in a frame alters the projection of the frame onto the eigenspace. Although we might want to consider the sequence in Figure 4-2 one shot, the classifier attempts to cluster this sequence spatially. Since the background and foreground are dynamic, what it perceives as the background is altered.

4.1.1 Cluster Characteristics

There are some common characteristics that we can note about the framework of the clustering algorithm. It will detect fast dissolves, changes in location, and the presence of a different character. Figure 4-3 depicts a case where the clustering is

Figure 4-2: Example of Clustering with a Pan Camera Motion

in form with our perception of shots. Each of these frames has a designated cluster label. In this case it was {...13, 93, 17, 17, 95, 95, ...}. A disparity in the regular succession of classes designates a shot change.

We saw in Figure 4-2,that the intensity patterns occupying discrete areas of the frame heavily influence the classification. Figure 4-4 depicts a similar clustering. In this case, the features are altered by the change in the posture of the character. Specifically, the boy is rotating his head downwards. The classifier accepts this rotation within the bounds of a threshold. Once this threshold is surpassed, however, another cluster may lay closer in the eigenspace as it does in this case.

The classifier extracts the spatial domain of the movie. In doing so, it defines the principal elements of the movie. We have seen above how these shots differ from shots as defined in the film sense. The next step is to combine these elements and form a new movie.

4.2 Visual Grouping of Clusters

This system clusters movies. The viewing of all the frames and their respective labels poses a problem in terms of its organization. Ron MacNeil at the Media Lab's Visual Language Workshop has developed a *videograph* to explore the temporal development of action in a stream. We will use such a device to explore the rhythm of a movie with respect to its clusters.

4.2.1 Slit Scan View

A slit scan view views the same portion of each frame in a continuous strip where the slices are viewed with increasing time. In this particular case, we will view slits made up of the middle four pixels of each row of each frame (see Figure 4-5).

The entire movie is displayed in this strip. Since this strip is so long, the portion being viewed at any single time in the porthole is controlled by the scrollbar. In this

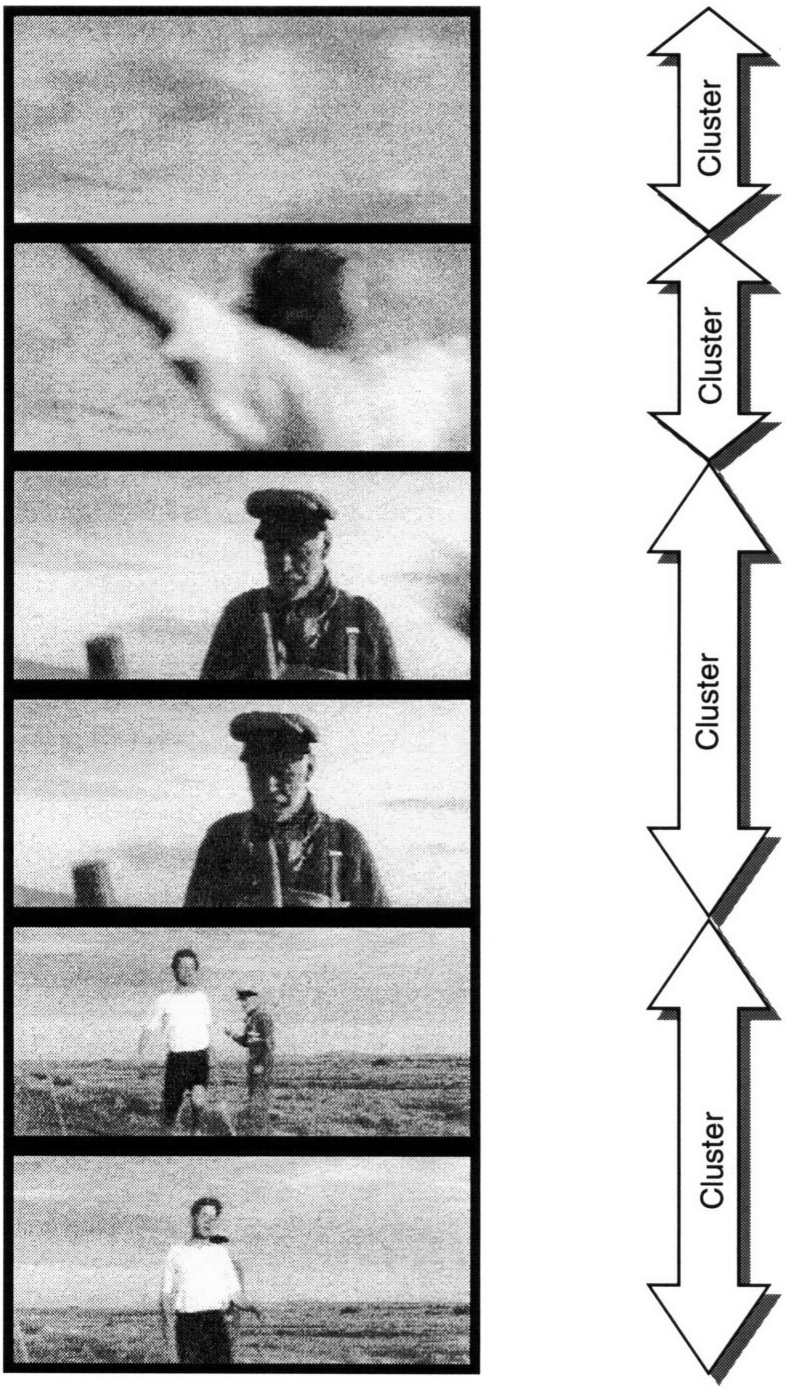


Figure 4-3: Sequence of Four Clusters

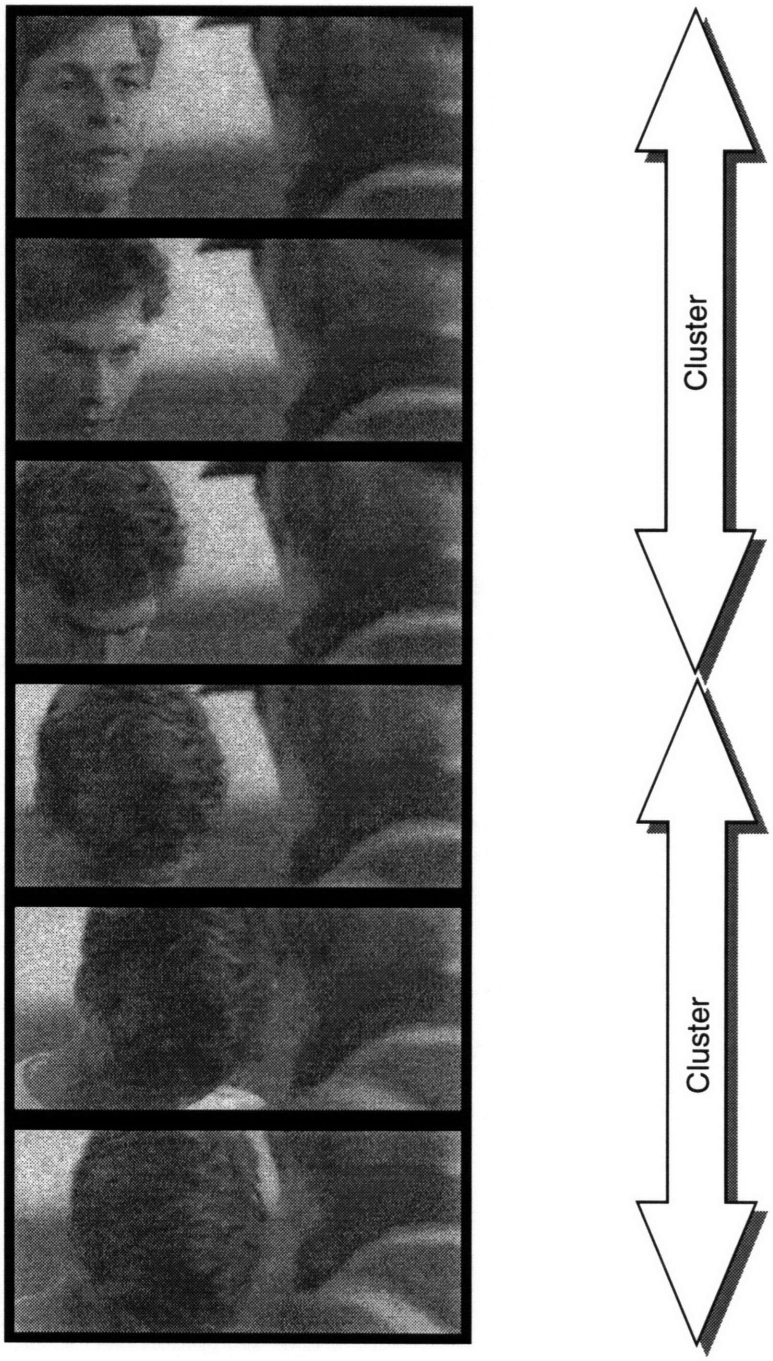


Figure 4-4: Example of Clustering with a Tilting Head

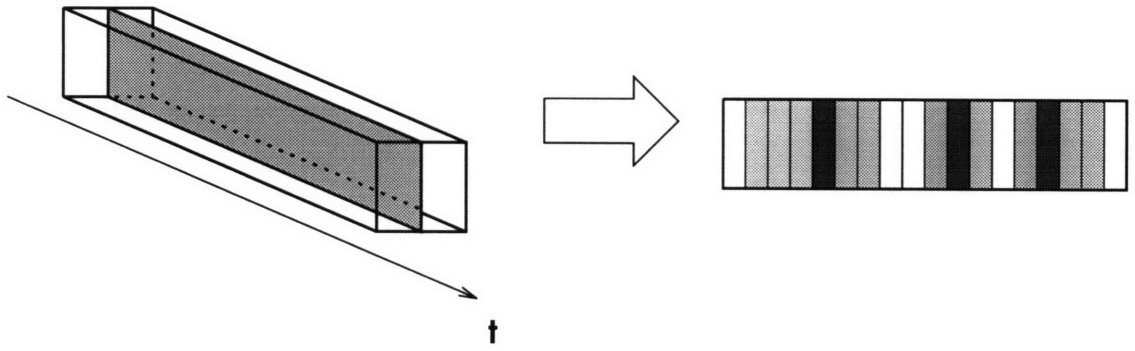


Figure 4-5: Slit Scan Model

implementation, a colored bar representing the cluster label is placed above the frame slices.

Figure 4-6 shows two sample views of this approach. The sharp transitions in the slices are also good indications of shot cuts. The first of the two pictures in Figure 4-6 begins with some very long shots. The slices display a continuity of form. The cluster bars above the slices are also longer in this portion of the strip. The bottom picture has very jagged and abrupt transitions in the splices and many short cluster bars. This is an indication of rapid movement or an action sequence. We can use this clustering information then to find how the clusters relate to the rhythm and flow of the movie.

4.2.2 Integrating Clusters using Salient Stills

In Section 2.4, we described a method for combining multiple frames into one frame using affine transformations. We will use that method here to reassemble our movie based on the designated clusters.

The clustering method creates elements which are affine. Once the clusters are made, the use of Salient Stills will produce relatively accurate representation of the group. Figure 4-7 shows the sequence from Figure 4-3 converted into a *salient* sequence.

Figure 4-6: Slit Scan Views of a Movie with Clustering

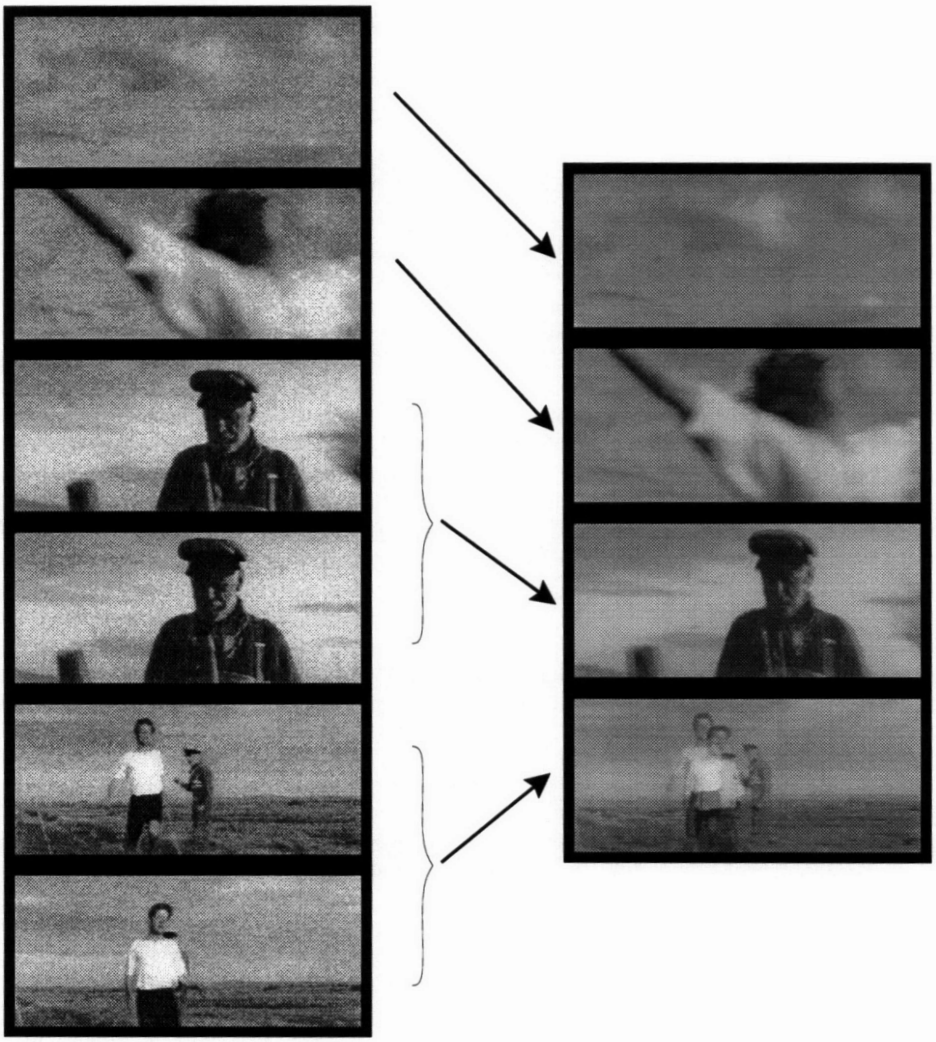


Figure 4-7: Creation of Salient Movie Stream

Chapter 5

Salient Stills for Salient Movies

5.1 Composition of Salient Movies

In Section 4.2.2 we began to show how a Salient Movie is created. In this chapter we will show some examples of salient sequences.

Figure 5-1 shows the opening sequence of *Terminator 2*. The stills of the movie were placed over the intervals of the frames they represent. We can see here how the redundancy in frames is captured. A Salient Movie would consists entirely of stills placed in a predefined order. They can be placed sequentially on a display. Figure 5-2 contains a description of a section of the movie *Gallipoli*.

The system succeeds in differentiating different characters and different surroundings. As noted earlier however, slight rotations or a reorganization of the allocated frame space will result in a new element.

5.2 Non-Traditional Form of Viewing

The resulting frame from a Salient Still does not always have the same dimensions as the original frames. When a camera zooms in or out, the final picture is larger. Figure 5-3 is a collage of stills from a movie stream. The large pictures originated

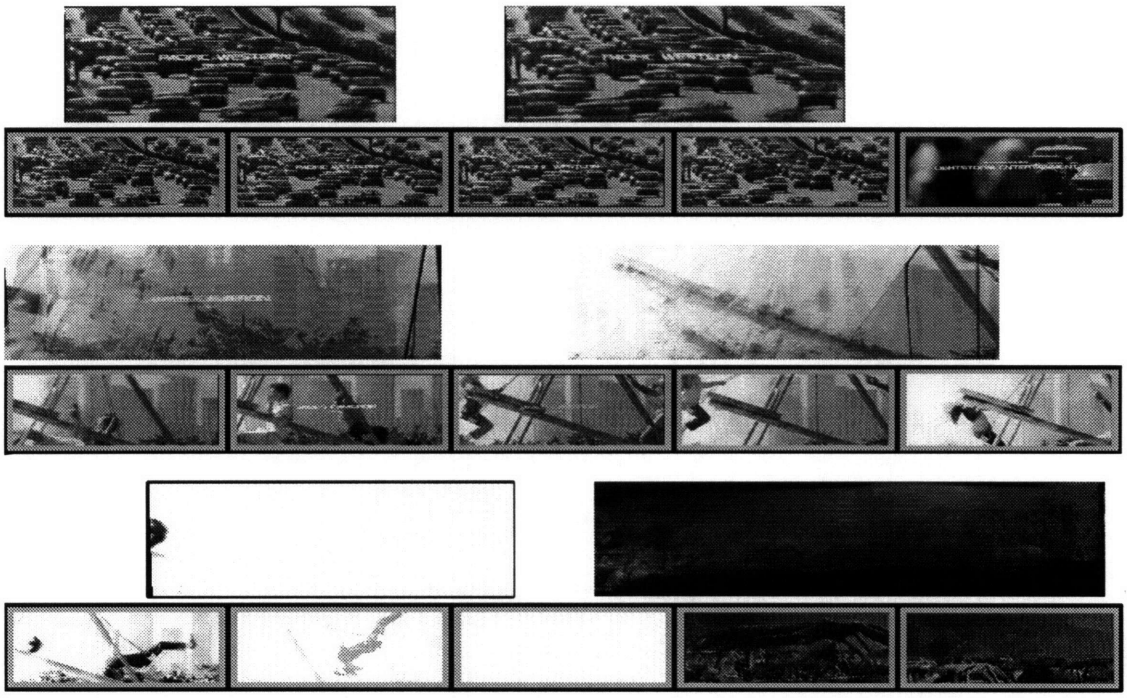


Figure 5-1: Beginning of a Salient Movie

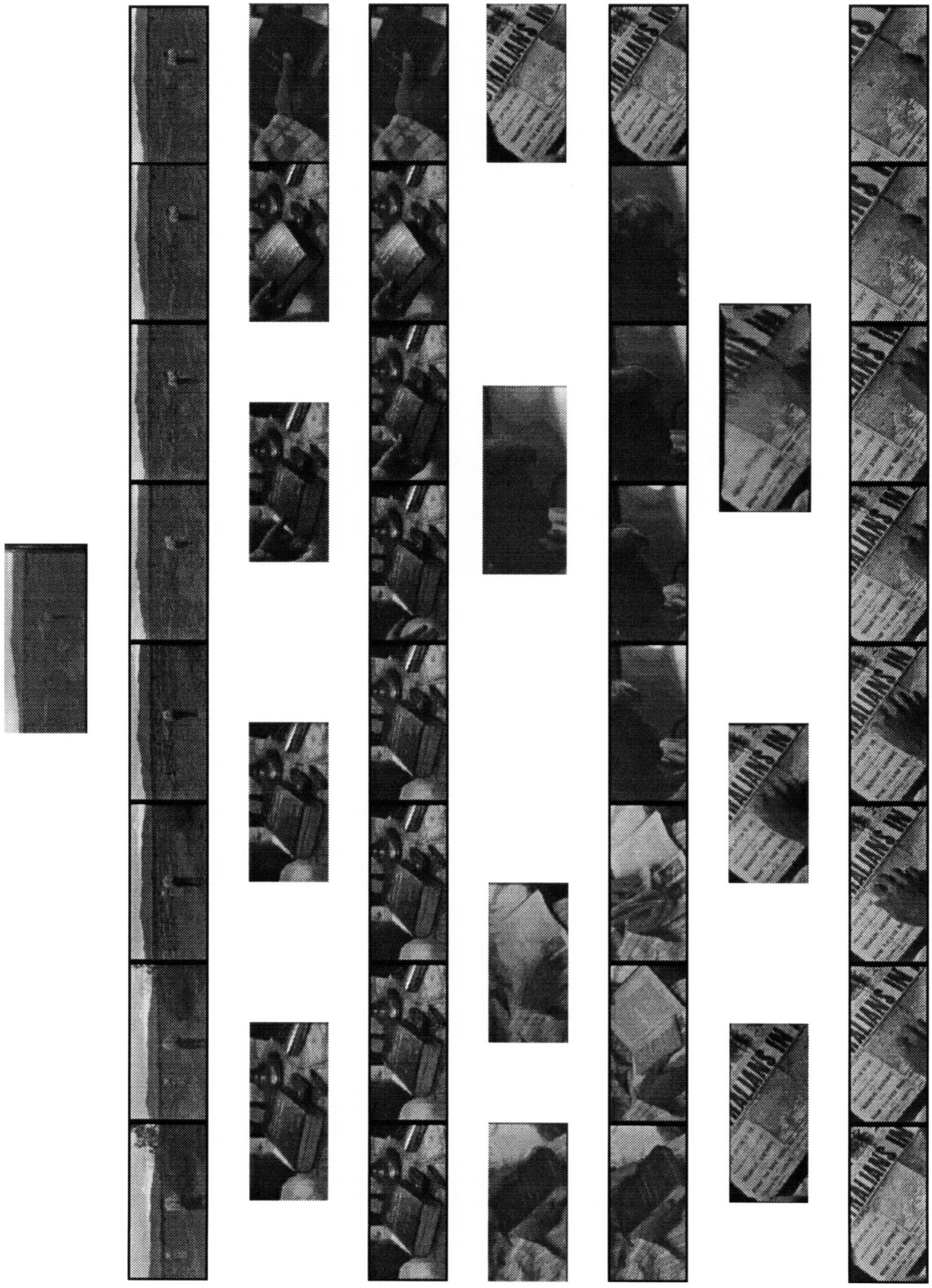


Figure 5-2: Stream from the movie *Gallipoli*

from zooms. The sequential order of the pictures is counter clockwise from the upper left.

The first still originated from a sequence of Indians running in and out of a tepee. The still filtered out the motion and maintained the background of the scene. The third frame is a zoom of a house, followed by a zoom of the characters in that house.

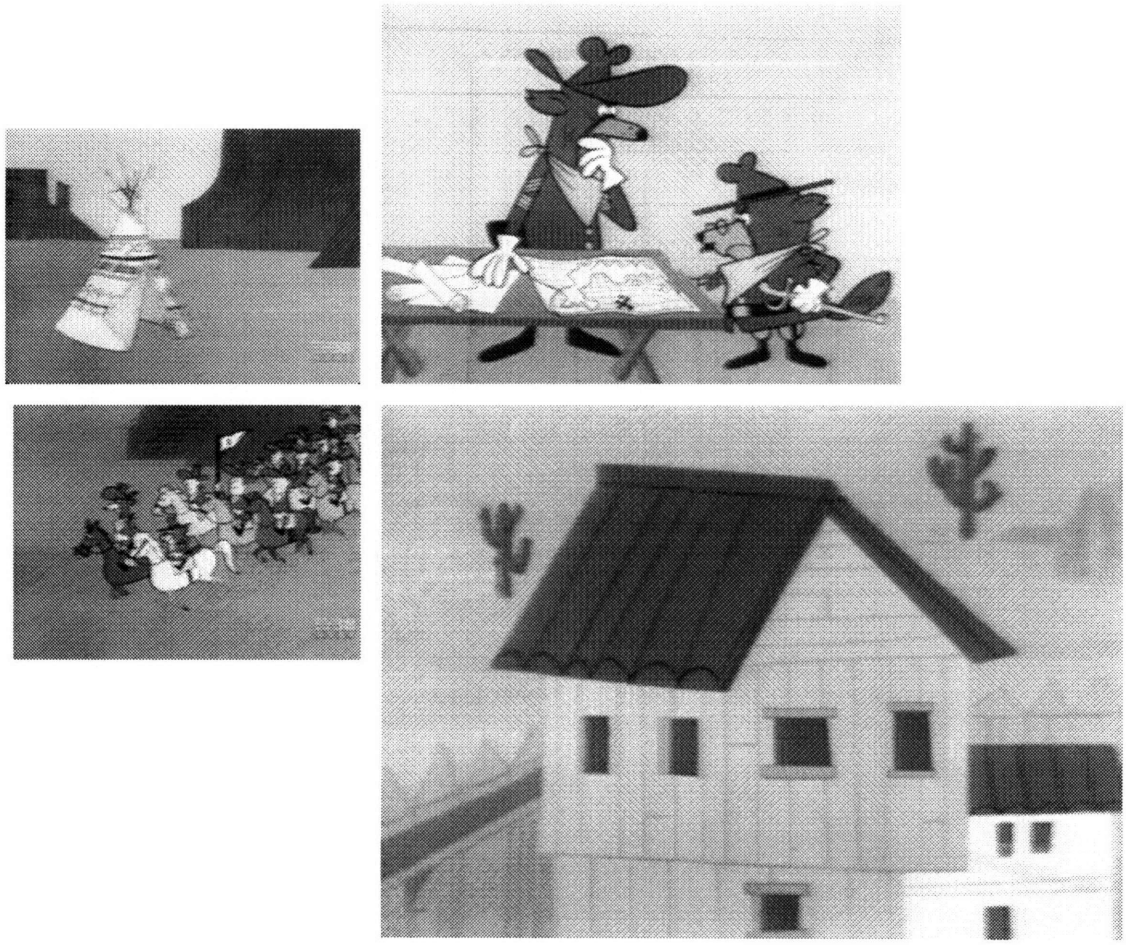


Figure 5-3: Two-dimensional Salient Collage

5.3 General Observations

The pace of a movie is captured by it's Salient Movie counterpart. An action film would be comprised of many clusters representing the various focuses of attention. A slow moving movie would be characterized by large clusters. In this case, the objects occupying the frame area, would not be significantly changing, and we can assume a lack of action.

The resulting Salient Movie is non-linear in time. It focuses on the complex rhythmic and dynamic continuity of space. It depicts much of the physical environment of the sequence and changes in camera motion as well as camera angles. One salient frame contains more information with regards to the content of a chunk of the movie than does a single frame in that cluster.

Chapter 6

Future Work and Relevant Extensions

All of the techniques used in making Salient Movies are open ended and quite flexible. Therefore many variations on the theme can be made.

6.1 Color

All the computations in the system involved only the luminance values of the pixels. Using the available color information will ultimately lead to a trade-off between increased accuracy and increased complexity in solving the covariance matrices.

6.2 Motion Processing

Principal component analysis tended to segment pans and rotations into more than one cluster. This is due to the shift invariance.

A pan is a continuous event with a dynamic background. Therefore, given that the frames in the movie are aligned before the classification process, the entire pan would be in one cluster. Alignment would compensate for camera motion and also for

multiple moving objects. This would allow for larger clusters representing full shot sequences. This can be done using a method similar to the optical flow estimation used in Salient Stills.

6.3 Teaching the System

This would require modifying the block diagram of Figure 3-2 and would allow the system to alter the structure of the clusters[11]. If a frame's projection was not near any of the known classes, it would create a new class. Conversely, the number of clusters can also be decreased if two clusters can function as one.

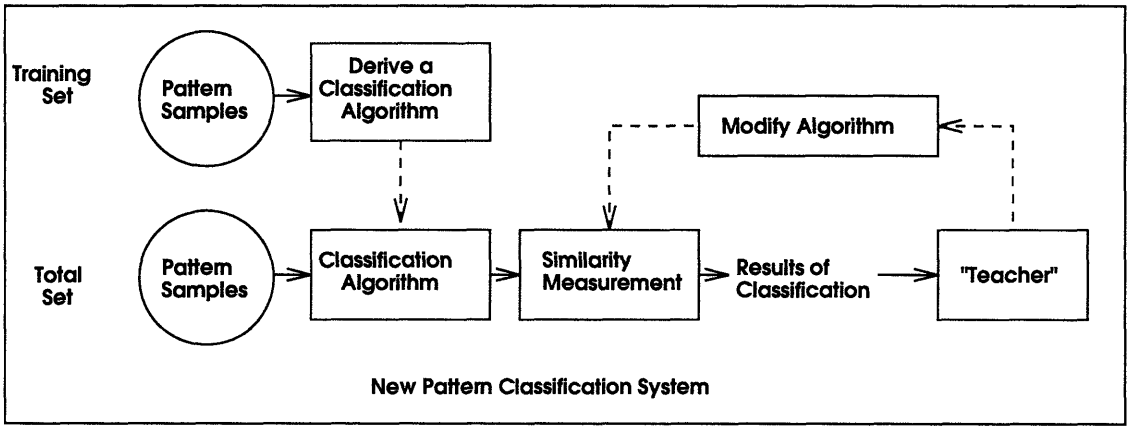


Figure 6-1: Modified Block Diagram of the Classification System

6.3.1 Neural Networks

Whenever learning is involved, neural nets are soon to follow. This system can be implemented using a collection of neural networks. One could be responsible for the projection onto the eigenspace and another for the classification [12].

Chapter 7

Conclusion

Salient Movies is a tool. It performs the task of transforming time into space. The main contribution of this thesis is the design and implementation of a new viewing form for movies and other film sequences. The approach used reveals many concepts in the understanding of video:

- The manner in which a story is advanced before the viewer.
- The continuous mental reconstruction of the sequence.
- A visualization of the rhythm and flow of a movie sequence.
- The perception of time and space.

This new representation of the movie retains the content of the movie in a condensed form suitable for browsing. This is only one way of visualizing the *fabric* of a movie. Still, it succeeds in seeking the key elements of a movie and displaying its motion in a two-dimensional mosaic.

Appendix A

Estimation of the Affine Parameters

We model optical flow as a continuous variation of intensity[10, 7, 1].

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{A.1}$$

Because we are assuming that the intensity is continuous, we can make a Taylor series expansion of the right hand side of Equation A.1.

$$\frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} = -\frac{\partial I}{\partial t} \tag{A.2}$$

With Equation A.2, we can model pans and tilt, for they depend on the horizontal and vertical components only. To include zooms, the following six parameter model is used (see Section 2.4.1).

$$v_x(x, y) = a_x + b_x x + c_x y \tag{A.3}$$

$$v_y(x, y) = a_y + b_y x + c_y y \tag{A.4}$$

The goal is to determine the values for a_x, b_x, c_x, a_y, b_y and c_y. This involves finding

a best fit solution for two sets of data onto these two models. Least squares is used for the fitting of the plane to the motion vectors v_x and v_y:

$$\begin{aligned} LSE &= \sum_R (v - (a + bx + cy))^2 \\ &= \sum_R v^2 - 2v(a + bx + cy) + (a + bx + cy)^2 \end{aligned} \tag{A.5}$$

where R is the region of the image used in the estimation.

We want to find the parameters that minimize the least square error. This corresponds to differentiating Equation A.5 with respect to each of the affine parameters. The resultant equations are:

$$\begin{aligned} a\sum_R 1 + b\sum_R x + c\sum_R y - \sum_R v(x,y) = 0 \\ a\sum_R x + b\sum_R x^2 + c\sum_R xy - \sum_R xv(x,y) = 0 \\ a\sum_R y + b\sum_R xy + c\sum_R y^2 - \sum_R yv(x,y) = 0 \end{aligned} \tag{A.6}$$

These equations can be represented in the matrix form as follows:

$$\begin{bmatrix} \Sigma_R 1 & \Sigma_R x & \Sigma_R y \\ \Sigma_R x & \Sigma_R x^2 & \Sigma_R xy \\ \Sigma_R y & \Sigma_R xy & \Sigma_R y^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma_R v(x,y) \\ \Sigma_R xv(x,y) \\ \Sigma_R yv(x,y) \end{bmatrix} \tag{A.7}$$

This is a case of three equations and three unknowns. Solving this matrix give us the unknown affine parameters.

Bibliography

[1] James R. Bergen, Peter J. Burt, Rajesh Hingorani, and Shmuel Peleg. Computing two motions from three frames. *David Sarnoff Research*, April 1990.

[2] Inc Dubner International, May 1995. Telephone conversation with Mr. Bob Dubner.

[3] Edward Lee Elliott. Thinking with motion images via streams and collages. Master's thesis, Massachusetts Institute of Technology, 1992.

[4] A. Hampapapur, R. Jain, and T. Weymouth. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1(1), March 1995.

[5] Berthold Klaus Paul Horn. *Robot Vision*. The MIT Press, 1986.

[6] Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 1988.

[7] Roger George Kermode. Building the big picture: Enhanced resolution from coding. Master's thesis, Massachusetts Institute of Technology, 1994.

[8] A. Pentland, T. Starner, N. Etcoff, N. Masiou, O. Oliyide, and M. Turk. Experiments with eigenfaces, mit media laboratory. *Perceptual Computing Technical Report*, 194, August 1992.

[9] Martin Szummer, May 1995. Extensive personal contact with M. Szummer regarding scene cut techniques.

[10] Laura Tedosio. Salient stills. Master's thesis, Massachusetts Institute of Technology, 1984.

[11] Charles W. Therrien. *Decision Estimation and Classification.* John Wiley & Sons, 1989.

[12] Matthew Alan Turk. *Interactive-Time Vision: Face Recognition as a Visual Behavior.* PhD thesis, Massachusetts Institute of Technology, June 1991.

[13] Matthew Alan Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[14] Hongjiang Zhang, Chien Yong Low, and Stephen W. Smoliar. Video parsing and browsing using compressed data. *Multimedia Tools and Applications*, 1(1), March 1995.